e-credit Reference Manual

Loans Online

Reference Manual for the Integration of UAB "Artea Lizingas" e-credit System within the Existing Web Shop: A Customer-Oriented Solution for Enabling Deferred Payments for Selected Services or Goods. Programming Guidelines.

Artea

2026/10/22

Processing modes

There are 2 processing modes available. The major differences are as follows:

Processing Mode	1	3
Mandatory digital signature of personal data use consent	yes	yes
The terms are selected, and the contract is concluded by	seller	client
Prepayment required by partner	yes	yes
Prepayment required by e-credit	no	yes
Status code returned	11/10	31/30

Processing mode (PM) is being selected by the partner by passing the corresponding value of the request XML *ReturnType* node.

E-shop request

The e-credit is launched from the partner's website by passing the request XML. The POST request for the final version of the application should be initiated on https://ecredit-api.artea.lt/UBLOnline.aspx?eshopdata=? while the same for the test version applies on https://ecredit-api-demo.artea.lt/UBLOnline.aspx?eshopdata=?
The XSD schema used for the request validation can be found at https://e-credit.sblizingas.lt/eshop/contractrequest.xsd

e-credit response

e-credit responds with POST request on the URL specified in the request XML *ReturnUrlStatus* node adding the parameters described below. The parameters specified in the request XML *ReturnUrlStatus* node can be accessed both either by querying POST or GET parameters.
In case the parameters specified by the partner overlap, they can only be accessed using the GET query.
The request XML *ReturnUrl* node is used for redirecting the customer only (both in case the process has been terminated of fully completed) – no response is sent to this URL.

Parameters

status

Parameter *status* is used to indicate application processing outcome. Possible values are:

Value	Description
10	**the consent to use personal data missing** – application rejected – returned in case of the 1st processing mode (PM) only
11	**the consent to use personal data digitally signed** – free to prepare the contract draft and deliver it for signing – returned in case of the 1st PM only
30	**application rejected** – returned in case of 3rd PM only
31	**the contract is digitally signed** – wait for the payment – returned in case of 3rd PM

The response to application rejection is not immediate. In case of fully automated processing, it will be available within 2 hours. Otherwise, it will be available in 24 hours in most cases, except those where the additional papers would be required from the client. The partner is therefore always advised to implement his own timeout procedure on top of the available solution.

orderID

Parameter *orderID* contains application internal ID in e-credit.

orderNr

Parameter *orderNr* contains application ID in case *status* is 10 or 11, or contract ID otherwise.

paymentDue

Parameter *paymentDue* contains the payment due to the partner in the requested currency.

granted

Parameter *granted* contains the credit principal in the requested currency.

ts

Parameter *ts* contains the response timestamp.

control

Parameter *control* is used to secure the integrity of the transaction (deprecated)

control2

Parameter *control2* is used to secure the integrity of the transaction. It's the string of all the above parameters, digitally signed using e-credit private key, using the SHA512 signing algorithm. The signature is Base64 encoded.

The parameter string must be created as follows (replace ? for the actual parameter values):
"status=?&orderID=?&orderNr=?&paymentDue=?&granted=?&ts=?&returnUrlStatus =?&saleLogin=? "

Please, make sure of the order of the parameters in the parameter string above. It cannot be modified. All the parameters must be present.

The partners using the 3rd PM must verify the signature using the public key available as a part of the *certificate* on the receipt of *status* 31. In addition, they are obliged to store all the response parameters in their databases. PHP and C# sample code for the signature verification is available as an appendix to this document.

Signed responses must be always stored in the partner's database, including all parameters and the signature itself. This serves as the sole proof that a transaction has occurred, which can be used in case of arbitration.

Loan amount

The shopping cart total PKS (provided by the request XML) = prepayment requested by the partner API (always >=0) + credit requested CT (provided by the request XML).
Credit requested CT (provided by the request XML) = payment due to the partner PAY (provided via the parameter *paymentDue*) + prepayment requested by e-credit, but collected by the partner APC (always >=0).
Payment due to the partner PAY (provided via the parameter *paymentDue*) = credit granted CG (provided via the parameter *granted*) + prepayment requested and collected by e-credit AIC (always >=0).

The link between the variables and the constraints is as follows:

$$PKS = \underbrace{\underbrace{A_{IC} + CG}_{PAY} + A_{PC} + A_{PI}}_{CT}; A_{IC} \geq 0; A_{PI} \geq 0; A_{IC} \cdot A_{PC} = 0$$

⚠ In case of 3rd PM, UAB "Artea lizingas" financing may be partial to the one requested. The partner is always expected to verify the paymentDue response parameter, i.e., the actual money transferred in case the contract is signed. In case it does not cover the shopping cart the payment should be deemed as partial, and the rest of the settlement should be done independently.

Sample XML

Standard request

The fully functional XML request sample is presented below. Please, replace *demo* in *SaleLogin* node with your own login name. This XML contains all the possible configuration options. The required configuration options are listed in the minimal request.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ContractRequest
    xmlns="http://e-credit.ubl.lt/eshop"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://e-credit.ubl.lt/eshop http://e-
credit.sblizingas.lt/eshop/contractrequest.xsd">
    <!--Connection details-->
    <SaleLogin>demo</SaleLogin>
    <!--Artea lizingas partner login-->
    <SaleConditionID>0</SaleConditionID>
    <!--Artea lizingas terms applied, if any-->

    <Person> <!--Customer data-->
        <FirstName>Jonas</FirstName>
        <!--Customer or appointed person name-->
        <LastName>Jonaitis</LastName>
        <!--Customer or appointed person surname-->
        <Code>30202020000</Code>
        <!--Customer or appointed person personal identification number-->
    </Person>

    <Communication> <!--Customer addresses-->
        <StreetOrVillage>Joninių g.</StreetOrVillage>
        <!--Name of the street or the village-->
        <BuildingNumber>1</BuildingNumber>
        <!--Building number-->
        <ApartmentNumber>1a</ApartmentNumber>
        <!--Flat Number, if any-->
        <City>Jonava</City>
        <!--Name of the city, the town or the region-->
        <PostCode>20000</PostCode>
        <!--Postal code-->
        <Phone>37061111111</Phone>
        <!--Mobile phone number-->
        <Email>jonas@jonaitis.lt</Email>
```

```
        <!--Email-->
    </Communication>

    <ContractDetails> <!--Contract terms requested by the customer-->
        <Term>12</Term>
        <!--Term in months-->
        <PaymentDay>30</PaymentDay>
        <!--Payment execution day-->
        <CreditAmount>1000,00</CreditAmount>
        <!--Loan amount requested. Use either comma or full stop separator-->
        <CreditCurrency>EUR</CreditCurrency>
        <!--Loan currency according to ISO 4217(EUR)-->
    </ContractDetails>

    <OrderedItemsDetail> <!--Product cart and additional info-->
        <Item> <!--Item description-->
            <Name>Unit</Name>
            <!--Item name-->
            <Amount>2</Amount>
            <!--Number of units-->
            <Price>300,00</Price>
            <!--Item code-->
            <Code>ABC</Code>
            <!--Item price: use either comma or full stop separator-->
            <Currency>EUR</Currency>
            <!--Currency according to ISO 4217(EUR)-->
        </Item>
        <Item> <!--Item description-->
            <Name>Service</Name>
            <!--Item name-->
            <Amount>1</Amount>
            <!--Number of units-->
            <Price>400,00</Price>
            <!--Item code-->
            <Code>ABC</Code>
            <!--Item price: use either comma or full stop separator-->
            <Currency>EUR</Currency>
            <!--Currency according to ISO 4217(EUR)-->
        </Item>
    </OrderedItemsDetail>

    <!--Important information-->
    <ShopLogoUrl> https://cdn.artea.lt/mainweb/logo.svg</ShopLogoUrl>
    <!-- Partner logo url-->
    <ReturnUrl>http://www.artea.lt</ReturnUrl>
    <!--Customer return address-->
    <ReturnUrlStatus>https://www.artea.lt</ReturnUrlStatus>
    <!--Additional code (Used as an additional order identifier, optional)-->
    <AdditionalCode>ABC</AdditionalCode>
    <!--Additional account no. (Used as an additional order identifier, optional)-->
    <AdditionalAccountNo>ABC</AdditionalAccountNo>
    <!--Response delivery address-->
    <ReturnType>3</ReturnType>
    <!--Processing mode-->
</ContractRequest>
```

Minimal request

The fully functional XML request sample is presented below. Please, replace *demo* in *SaleLogin* node with your own login name. This XML contains only the required configuration options. Please, refer to standard request for the additional configuration options. Using them would greatly increase your customer user experience.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ContractRequest
    xmlns="http://e-credit.ubl.lt/eshop"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://e-credit.ubl.lt/eshop http://e-
credit.sblizingas.lt/eshop/contractrequest.xsd">

    <!--Connection details-->
    <SaleLogin>demo</SaleLogin>
    <!--Artea lizingas partner login-->

    <ContractDetails> <!--Contract terms requested by the customer-->
        <CreditAmount>300,00</CreditAmount>
        <!--Loan amount requested. Use either comma or full stop separator-->
        <CreditCurrency>EUR</CreditCurrency>
        <!--Loan currency according to ISO 4217(EUR)-->
    </ContractDetails>

    <OrderedItemsDetail> <!--Product cart and additional info-->
        <Item> <!--Item description-->
            <Name>Unit</Name>
            <!--Item name-->
            <Amount>1</Amount>
            <!--Number of units-->
            <Price>300,00</Price>
            <!--Item price: use either comma or full stop separator-->
            <Currency>EUR</Currency>
            <!--Currency according to ISO 4217(EUR)-->
        </Item>
    </OrderedItemsDetail>

    <!--Important information-->
    <ReturnUrl>https://www.artea.lt</ReturnUrl>
    <!--Customer return address-->
    <ReturnType>3</ReturnType>
    <!--Processing mode-->
</ContractRequest>
```

Signature verification sample code

C#

The sample function used to verify the signature is presented below.

```
using System;
using System.Data;
using System.Text;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;


bool TestSignature(string CertFileName, string Signature, string TestString)
{
    bool success = true;
    try
    {
        X509Certificate2 cert = new X509Certificate2(CertFileName);
        SHA512Managed sha512 = new SHA512Managed();
        RSA csp = cert.GetRSAPublicKey();
        byte[] data = Encoding.UTF8.GetBytes(TestString);
        byte[] hash = sha512.ComputeHash(data);
        byte[] baseDecoded = Convert.FromBase64String(Signature);
        Array.Reverse(baseDecoded);
        success = csp.VerifyHash(hash, baseDecoded, HashAlgorithmName.SHA512,
RSASignaturePadding.Pkcs1);
    }
    catch { success = false; }
    return success;
}
```

PHP

The sample function used to verify the signature is presented below.

```
function testSignature($CertFileName, $Signature, $TestString)
{
    $success = true;
    try {
        $fp = fopen($CertFileName, "r");
        $cert = fread($fp, 8192);
        fclose($fp);
        $pubKey = openssl_get_publickey($cert);
        if (openssl_verify($TestString, strrev(base64_decode($Signature)), $pubKey,
OPENSSL_ALGO_SHA512) !== 1) {
            $success = false;
        }
        openssl_free_key($pubKey);
    }

    catch(Exception $e) {
        $success = false;
    }
    return $success;
}
```

Parameters

Parametras	Tipas	Aprašymas
CertFileName	string	Certificate file name
Signature	string	Signature string received through POST *control2* parameter
TestString	string	Well-formed and signed POST parameter string

Return Values

Tipas	Aprašymas
bool	**true** – on success, **false** – on failure

Frequently asked questions

Empty or incomplete response

Case

The partner website URL is invoked by e-credit, however no parameters like *status*, *orderID*, etc. appear to have been received.

Solution

Asynchronous responses are being sent to the URL specified by *ReturnUrlStatus* instead of *ReturnUrl*. In case the request XML does not contain *ReturnUrlStatus*, no response would be sent, while the partner most probably is expecting it on the URL specified by *ReturnUrl*. The latter is reserved for the customer redirection only – no responses are being sent to it. The following steps should be taken:

1. URI link specified in *ReturnUrl* node must be copied to *ReturnUrlStatus* node
2. Make sure that the URL specified in *ReturnUrl* is used as the next page the customer would land after completing the loan application. This page must return the status 200 OK to our servers and contain the actual content. No redirections to the URL specified in *ReturnUrl* are allowed, or e-credit will ignore them and fail to return the customer to the website

The response is being sent to the URI specified by the request XML *ReturnUrlStatus* node asynchronously and only once. The URI specified in the *ReturnUrl* is being triggered several times at least, even before the actual customer redirection, and in most cases before the response is being sent. No parameters are being passed to this URI.

How can I get e-credit certificate?

Case

A valid e-credit certificate is required to verify the signature; however, the certificate is not included with the message or document.

Solution

The e-credit certificate used for signature validation can be downloaded directly from the link: certificate

No response from e-credit

Case

Website does not receive application status responses on consent or contract signing.

Solution

Usually the responses on the consent/contract signature status are being sent within few minutes if the user is able to complete the application without an intervention. However, in case the user requires some assistance, or the information provided by the user is incomplete, the response mostly will be sent within the 24 hour period. Please just wait for missing responses if you receive at least some of them. However, if you do not receive any response, please make sure that:

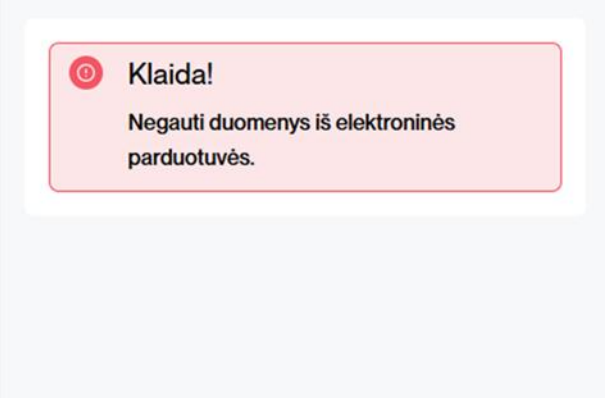
1. Request XML *ReturnUrlStatus* node contains the correct URI, listening for the response
2. Firewalls are not blocking the ports 80 and 443 on the URI specified by the *ReturnUrlStatus*. Easy way to check this can be found here – https://httpstatus.io/

⚠ Asynchronous responses are being sent to the URI specified in *ReturnUrlStatus* node instead of *ReturnUrl* node. The response will not be sent if the request XML contains only *ReturnUrl* node and no *ReturnUrlStatus* node.

Error on the landing page

Case

After clicking the e-credit link in the e-shop website, the user is redirected to e-credit site, however the site itself displays the error message, indicating that the application data has not been received.



Solution

The message is usually shown in one of those three cases:

1. Request XML *SaleLogin* node contains invalid login information
2. Request XML is not well-formed or it is invalid according to XSD schema
3. The loan amount requested as within the *CreditAmount* node exceeds the order product cart total price

Make sure that:

1. *OrderedItemsDetail* sub-nodes contain the full order basket. The total *Price* * *Amount* sum for all items must be equal or greater than the *CreditAmount*
2. XML is valid according to XSD schema - easy way to check it is using the following link http://www.utilities-online.info/xsdvalidation/

If the problem persists, most probably you are using incorrect login. This may happen while using test environment login on the release version and vice versa. It is also possible that the login has been disabled for some reason. You should contact “Artea” employee in this case.

Redirect to the partner website does Not Work

Case

The link to the partner's website is not working properly – the user is redirected to the main "Artea" site instead.

Solution

To maintain adequate user experience e- credit is automatically checking whether the link specified in the Request XML *ReturnUrl* node complies with the following rules:

1. Site exists, i.e., the server responds, and the response code is not *404 Not Found*
2. The user is not being redirected somewhere else, i.e., the server response HTTP code must be 200 OK, not 301, 302, 307. The tool to check it is available at https://httpstatus.io/
3. The site is empty or non-informative

The links to the sites which do not comply with the above requirements are automatically replaced with the link to the main "Artea" website. Please, make sure that the *ReturnUrl* node contains the correct link, which should be used to redirect the user when he finishes the application. It must not be the page performing some backend action, including the processing of e-credit responses (as it is triggered more than once). It must also meet the above requirements.

Is the bank payment necessary?

Case

The user is being asked to make 0.01 € payment while testing the integration in order to sign the consent and/or contract. Is there any workaround to bypass the bank transfer?

Solution

The application and the agreement can be signed by simulating a bank transfer using one of the following personal identification numbers in the application: 30202020000, 30303030004, 50001018865.
Alternatively, signing can be completed using Mobile-ID/Smart-ID by entering test credentials